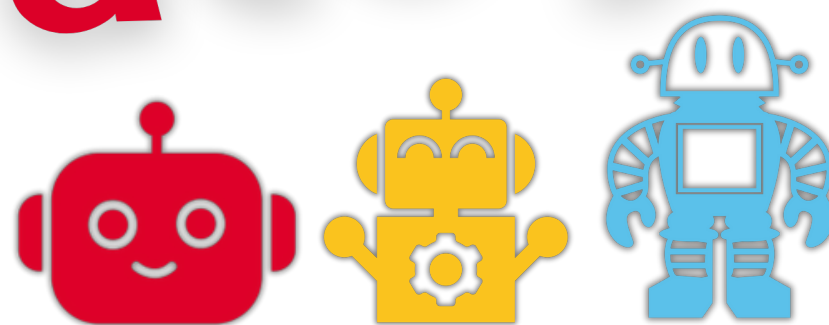


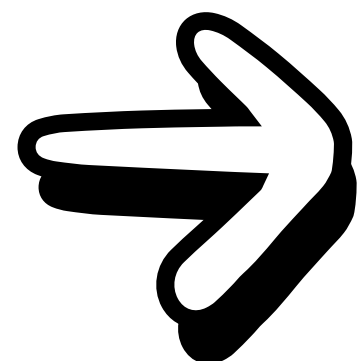
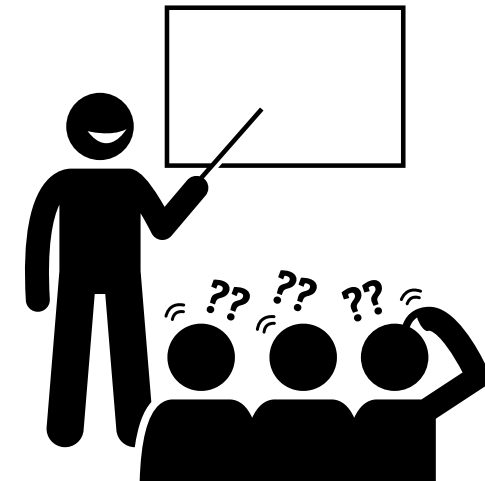
# METODOLOGÍA

## Estrategias

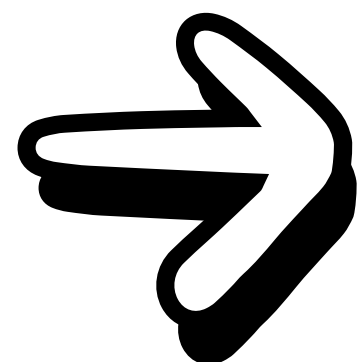




# ÍNDICE

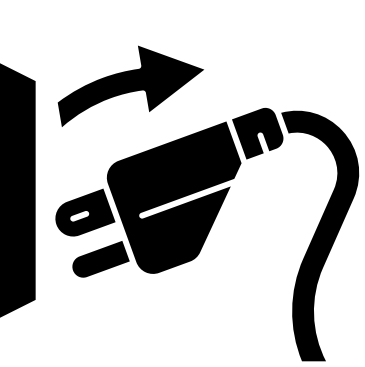


**1** Unificación de conceptos: pensamiento computacional, habilidades y actividad desenchufada.



**2** Línea metodológica para poner en práctica actividades de robótica y desenchufadas.





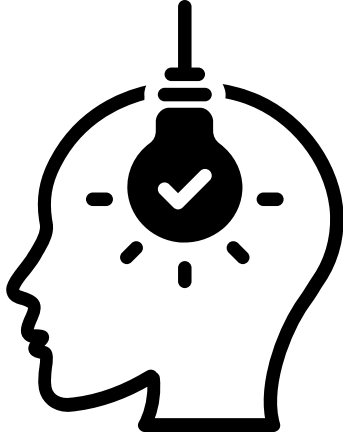
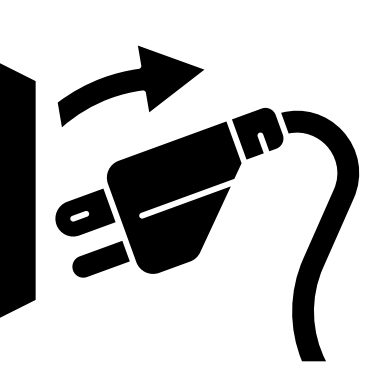
## DEFINICIÓN DE PENSAMIENTO COMPUTACIONAL

El pensamiento computacional es una forma de abordar problemas y diseñar soluciones de manera lógica y estructurada, similar a cómo lo haría una computadora.

Aquí tienes una definición sencilla y clara según varios autores:

- 1 Jeannette Wing (2006): Ella popularizó el término y lo define como una forma de pensar que implica resolver problemas, diseñar sistemas y comprender el comportamiento humano utilizando los conceptos fundamentales de la informática.
- 2 ISTE y CSTA (2011): Estas organizaciones educativas destacan que el pensamiento computacional incluye habilidades como la descomposición de problemas, el reconocimiento de patrones, la abstracción y el diseño de algoritmos.
- 3 García-Peñalvo (2016): Este autor resalta que el pensamiento computacional no solo es útil para la programación, sino también para desarrollar habilidades analíticas y de resolución de problemas en diversas áreas.

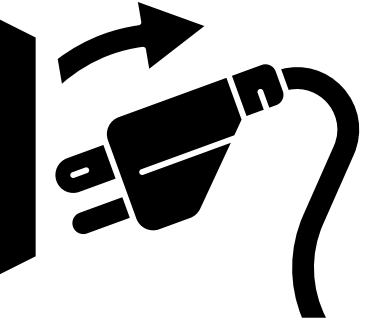




## Las habilidades asociadas al pensamiento computacional incluyen:

- **Lógica (predicción y análisis):** utilizar el razonamiento para hacer predicciones, resolver problemas y tomar decisiones basadas en la información disponible.
- **Algoritmos (pasos y reglas):** seguir una serie de pasos o instrucciones bien definidas para resolver un problema o completar una tarea.
- **Descomposición (dividir en partes):** dividir un problema grande en partes más pequeñas y manejables, que son más fáciles de entender y resolver.
- **Patrones (detectar y usar similitudes):** identificar similitudes o patrones en problemas o datos, lo que facilita encontrar soluciones más rápidas y eficientes.
- **Abstracción (eliminar detalles innecesarios):** Simplificar un problema eliminando detalles que no son importantes, para enfocarse en lo que es relevante y esencial.





# Ventajas de trabajar el pensamiento computacional en el aula desde la NEUROEDUCACIÓN



- Efecto primacía y recencia
- Motivación
- Emoción

## **FF.EE (Funciones ejecutivas):**

- Control inhibitorio
- Flexibilidad cognitiva
- Memoria de trabajo

## **FF.EE:**

- Planificación
- Razonamiento lógico
- Resolución de problemas
- Autorregulación
- Atención

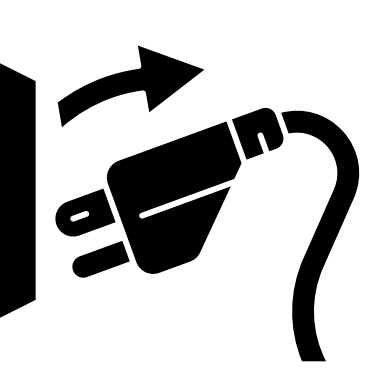
## **Además:**

- Creatividad
- Estructuración espacial
- Desarrollo psicomotricidad



Vídeo que plasma muy bien qué es el pensamiento computacional, cómo se puede aplicar a actividades cotidianas (hacer un sandwich).



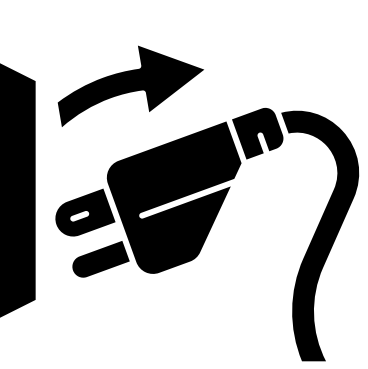


# Familiarización de los Docentes con Términos Técnicos

Es fundamental que los docentes se familiaricen con términos técnicos para mejorar su comprensión y enseñanza relativa al pensamiento computacional. Destacamos entre otros:

- **Variable:** "cajitas" donde guardamos información que puede cambiar o usarse más adelante.  
Ejemplo: crear una variable "puntos" en un juego para sumar puntos cada vez que un estudiante responda bien.
- **Secuencia:** son series de acciones desarrolladas en un orden específico, ya que se realiza una acción tras otra hasta que todas se hayan llevado a cabo. Las rutinas del día son secuencias.  
Ejemplo: seguir una receta.
- **Bucle:** es una instrucción que repite una acción varias veces.  
Ejemplo: saltar a la cuerda 10 veces. En programación, usarías un bucle para repetir la acción de saltar 10 veces.
- **Condicional:** es una instrucción que hace que el programa tome decisiones basadas en ciertas condiciones.  
Ejemplo: el juego: "¿Qué me pongo?". Los estudiantes aprenderán a usar condicionales para decidir qué ropa ponerse según el clima."





# Familiarización de los Docentes con Términos Técnicos

- **Evento:** acción que inician algo en el programa, como pulsar un botón o hacer clic en la pantalla.

Ejemplo: al presionar el botón "comenzar", se inicia un juego de preguntas y respuestas.

- **Función:** grupo de instrucciones que realiza una tarea específica, que pueden reutilizarse sin escribir todo de nuevo.

Ejemplo: hacer una función llamada "multiplicar" que multiplique un número por 2 y devuelva el resultado. Por ejemplo, si la función se llama con el número 4, dirá "El resultado es 8" cada vez que se use.

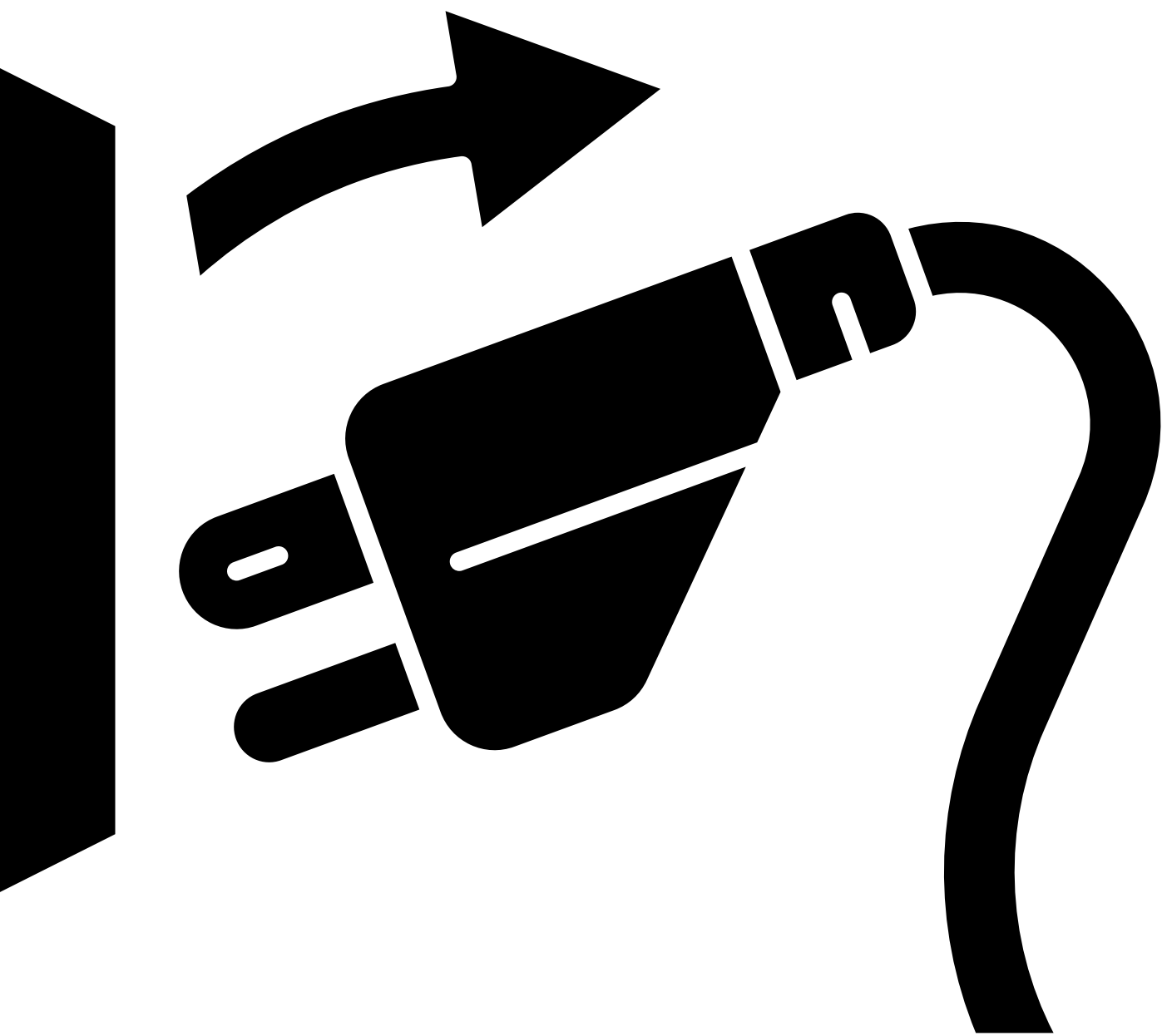
- **Patrón:** es una solución general y reutilizable para un problema común que aparece en diferentes contextos.

Ejemplo: el estudiante construye una torre con bloques de colores y quiere que siga un patrón específico: rojo, azul, rojo, azul. Este patrón te dice que después de un bloque rojo siempre debe ir un bloque azul y así sucesivamente.

- **Algoritmo:** es una lista de pasos ordenados que se siguen para resolver un problema o completar una tarea. Incluye condiciones y decisiones para manejar diferentes situaciones y llegar a una solución.

Ejemplo: hacer un algoritmo para preparar un bocadillo: "toma dos rebanadas de pan, pon mantequilla o salsa en una de las rebanadas, agrega algún tipo de embutido y queso, cubre con la otra rebanada de pan, corta el bocadillo por la mitad y sirve en un plato".

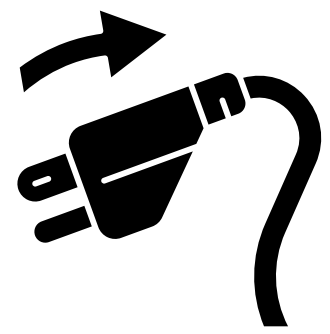




## DEFINICIÓN DE ACTIVIDAD DESENCHUFADA

Es una actividad educativa que se realiza **sin el uso de tecnología o dispositivos electrónicos**. Estas actividades están diseñadas para enseñar conceptos de pensamiento computacional y habilidades asociadas de manera divertida y accesible, utilizando materiales simples como papel, lápices, tarjetas, o incluso el propio cuerpo.

# 7 principios clave de las Actividades Desenchufadas



Usar tarjetas de instrucciones para simular la programación de un robot en un tablero de juego.

La actividad debe realizarse sin el uso de ordenadores, tabletas o robots reales. Se utilizan recursos manipulativos como tarjetas, papel, pizarras, tableros o elementos que representen comandos y conceptos de programa.

Los estudiantes resuelven un problema mediante pasos específicos y secuencias, identificando patrones que faciliten la solución.

La actividad debe fomentar habilidades de pensamiento computacional, como la división (dividir un problema en partes), la abstracción (simplificar información) y el reconocimiento de patrones.

Los estudiantes resuelven un problema mediante pasos específicos y secuencias, identificando patrones que faciliten la solución.

Las actividades desenchufadas suelen ser colaborativas, permitiendo el trabajo en equipos para resolver problemas, planificar estrategias o crear secuencias. Esto ayuda a desarrollar habilidades sociales y refuerza la comprensión a través de la discusión y el intercambio de ideas.

Los estudiantes resuelven un problema mediante pasos específicos y secuencias, identificando patrones que faciliten la solución.

Las actividades desenchufadas deben fomentar la creatividad y permitir que los estudiantes usen su imaginación. Además, es ideal que estén contextualizadas para que los conceptos se relacionen con situaciones del mundo real y sean más significativas para ellos.

1 **No dispositivos electrónicos**

3 **Pensamiento computacional**

5 **Trabajo en equipo**

7 **Creatividad y contexto**

2 **Programación robótica**

4 **Manipulación física**

6 **Accesibilidad y adaptabilidad**

Aunque no se utiliza tecnología, la actividad debe enfocarse en enseñar conceptos fundamentales de programación, robótica o pensamiento computacional, como algoritmos, bucles, condiciones y seguridad.

Crear un recorrido en el suelo y hacer que los estudiantes simulen ser robots, siguiendo instrucciones para llegar a una meta específica.

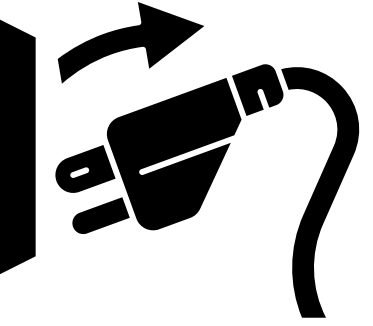
Las actividades diseñadas deben ser prácticas y físicas, permitiendo que los estudiantes manipulen objetos o realicen movimientos que representen conceptos de programación o robótica. Esto facilita la comprensión a través de la acción y la experimentación.

Usar una cuadrícula en el suelo y hacer que los estudiantes coloquen tarjetas de movimiento (adelante, gira a la derecha) para "programar" un recorrido.

Las actividades desenchufadas suelen ser más accesibles y fáciles de adaptar para estudiantes con diferentes niveles de habilidad o con necesidades educativas especiales, ya que no dependen de la tecnología. Pueden simplificarse o hacerse más complejas según las necesidades del grupo.

Ajustar la dificultad de la actividad al reducir o aumentar el número de pasos en una secuencia o al incluir opciones de retroalimentación y corrección.





# Línea metodológica para poner en práctica actividades de robótica y desenchufadas

1. Contexto de la actividad
2. Punto de partida/Consideraciones previas
3. Implementación en el aula
4. Desarrollo de la actividad
5. Producto final
6. Evaluación del producto y del proceso
7. Reflexión final para el docente



En las siguientes diapositivas explicamos cada uno de los pasos a seguir.





# 1. Contexto de la actividad

- Edad recomendada
- Nivel de dificultad
- Tipo de trabajo: individual, colaborativo, en grupos, por estaciones, etc.
- Narrativa: un aspecto a tener en cuenta es optar por enlazar varias actividades desenchufadas en torno a un contenido o proyecto específico.



# 2. Punto de partida/ Consideraciones previas

## 2.1. Competencias específicas.

## 2.2. Evaluación.

- Criterios de evaluación, debemos tener en cuenta desde el principio qué deben ser capaces de hacer nuestros estudiantes al finalizar la actividad (rúbrica definida).
- Evaluación del proceso:  
Participación y colaboración.  
Resolución de problemas y toma de decisiones.
- Autoevaluación y coevaluación (si se considera adecuada para el nivel).



## 3. Implementación en el aula

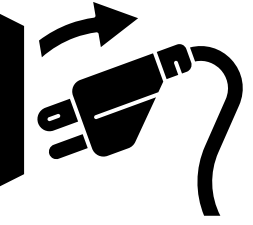


### 3.1 Preparación del aula

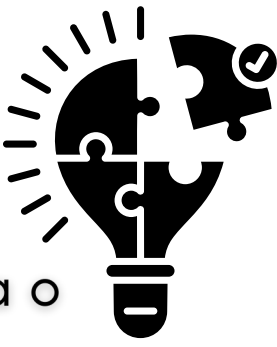
- Materiales: preparación de los mismos, que estén accesibles para los alumnos para favorecer la autonomía y adaptar aquellos que sean necesarios.
- Espacios de trabajo.
- Atención a las diferencias individuales del alumnado.
- Normas de uso y convivencia durante la actividad.

### 3.2 Activación

Presentación de la actividad. Recursos que podemos utilizar para ello (videos, presentaciones, material manipulativo, demostración por parte de un estudiante, plantear un problema a resolver sin mostrar ningún recurso, etc).



## 3.3 Metodologías a potenciar



- **Aprendizaje Basado en Proyectos (ABP)**

Descripción: los estudiantes trabajan en proyectos durante un período prolongado para investigar y responder a una pregunta, problema o desafío complejo.

Ejemplo: realizar un proyecto para diseñar y construir un robot. Investigan diferentes tipos de robots, planifican su diseño, construyen prototipos y presentan sus resultados.

- **Aprendizaje Basado en Juegos (ABJ)**

Descripción: utiliza juegos como herramienta educativa para motivar y enseñar conceptos específicos.

Ejemplo: programación con bloques (como ScratchJr) para resolver puzzles y desafíos. Cada nivel del juego introduce nuevos conceptos de programación de manera lúdica.

- **Gamificación**

Descripción: incorpora elementos de juego (puntos, niveles, recompensas) en actividades educativas para aumentar la motivación y el compromiso.

Ejemplo: crear una “liga de programación” donde los estudiantes ganan puntos y desbloquea niveles al completar tareas de programación y desafíos de robótica. Los puntos pueden canjearse por recompensas o privilegios en clase.

- **Paisajes de Aprendizaje**

Descripción: diseños educativos que integran diferentes actividades y recursos en un entorno de aprendizaje flexible y personalizado.

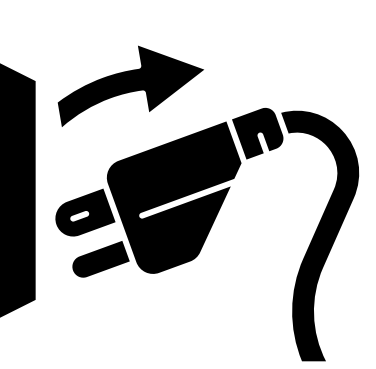
Ejemplo: los estudiantes pueden elegir entre varias actividades: construir un robot con LEGO, programar un robot para seguir una línea, o investigar sobre la historia de la robótica. Cada actividad está adaptada a diferentes niveles de habilidad y estilos de aprendizaje.

- **Estaciones de Aprendizaje**

Descripción: los estudiantes rotan entre diferentes estaciones cada una con una actividad o tarea específica (se recomienda especificar delimitar tiempo por estación)

Ejemplo: se presenta al grupo clase las siguientes estaciones: **1.** Construir un robot con bloques de construcción, **2.** Programación humana (moverse por un laberinto), **3.** Clasificar objetos por colores y **4.** Secuenciar imágenes de rutinas diarias.





# 4. Desarrollo de la actividad



## Tips para el Rol del Docente en la Fase de Desarrollo de la Actividad

### 4.1. Guía Socrática y Promoción del Pensamiento Crítico

Evitar respuestas  
directas  
Promover la deducción  
y la autorreflexión

### 4.2. Supervisión Estratégica del Grupo

Rondas de  
observación  
Análisis de  
interacciones

### 4.3. Evaluación Continua y Formativa

Registro de  
observaciones  
Feedback inmediato  
Autoevaluación

### 4.4. Refuerzo del Ambiente de Aprendizaje Positivo

Crear un ambiente  
seguro para el error  
Celebración de  
pequeños logros



# TRABAJO DEL ESTUDIANTE durante el desarrollo de la actividad



## Fase 1: Exploración o investigación

- Descripción y objetivos
- Materiales y recursos

## Fase 2: Diseño o planificación

- Descripción y objetivos
- Indicaciones generales


## Fase 3: Construcción o implementación

- Descripción y objetivos
- Indicaciones para guiar la actividad

## Fase 4: Verificación o retroalimentación

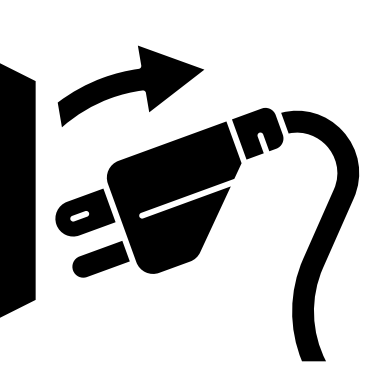
(puede incluir ensayo o pruebas de concepto)

- Descripción y objetivos
- Formato de revisión



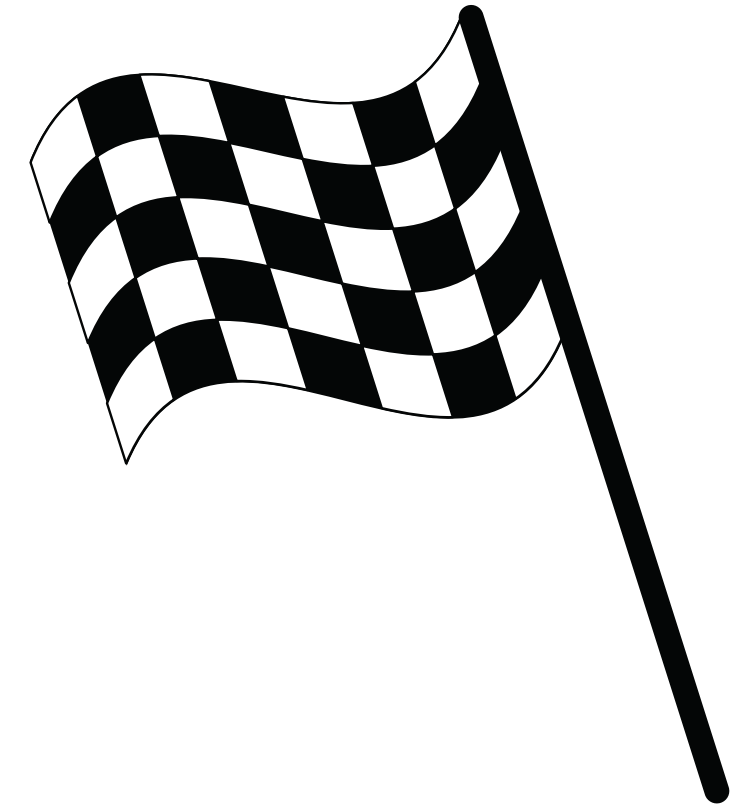
Las fases pueden adaptarse en número y secuencia según el tiempo y la profundidad de la actividad.

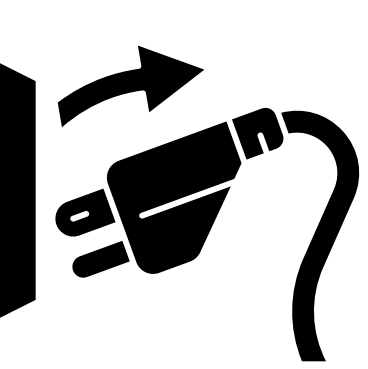




## 5. Producto final

- **Descripción del producto final esperado:** proyecto, dossier, ficha de trabajo, presentación, etc.)
- **Criterios de calidad o éxito del producto.**

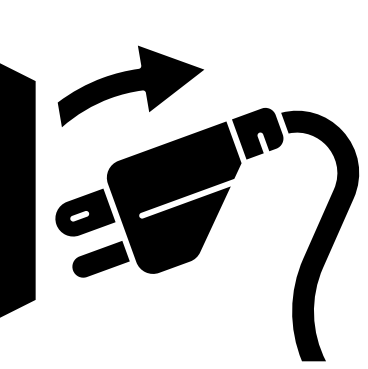




## 6. Evaluación del producto y del proceso (ya elaborado anteriormente)

- **Evaluación del producto final:**  
Criterios específicos (creatividad, funcionalidad, precisión, etc.).
- **Evaluación del proceso:**  
Participación y colaboración.  
Resolución de problemas y toma de decisiones.
- **Autoevaluación y coevaluación** (si se considera adecuada para el nivel).





## 7. Reflexión final para el docente

- Observaciones sobre el desempeño del grupo
- Desafíos encontrados
- Posibles mejoras para futuras implementaciones





*"Programa financiado por el Ministerio de Educación,  
Formación Profesional y Deportes"*

